Analysis of virtualization as a solution to VR-system sharing

Jean-Marie Normand, Aurélien Milliat, Marc Christie, Fabien Picarougne, Guillaume Moreau

Analysis of virtualization as a solution to VR-system sharing

Jean-Marie Normand*
CERMA, Ecole Centrale de Nantes

Aurélien Milliat†
LINA/CLARTÉ

Marc Christie‡
IRISA, University of Rennes 1

Fabien Picarougne§
LINA, University of Nantes

Guillaume Moreau¶
CERMA, Ecole Centrale de Nantes

ABSTRACT

Immersive Virtual Reality (VR) systems offer vast opportunities in terms of scientific research as well as in terms of industrial applications. However their construction and maintenance costs remain an issue. It is therefore essential to find means to easily and safely share VR equipments among multiple users in order to reduce such costs, as well as democratize their use.

In this paper, we analyze the interest of virtualization techniques to administrate, maintain and share complex VR systems among multiple users with heterogeneous needs in software, drivers, operating systems and peripherals. Despite the general idea that virtualization techniques strongly impact performance, we show that these technologies can be a viable solution with a careful selection of virtualization tools and graphics hardware.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality D.4.3 [Operating Systems]: File Systems Management—Distributed file systems

1 SHARING VR SYSTEMS

Although VR systems are becoming more and more popular, their cost remain high thus preventing small companies or research groups to use them. As a consequence, these systems should be shared among different users to at least cover construction, software and hardware maintenance and operator-related costs. In turn, this raises issues on how to effectively maintain multiple configurations for different users and how to efficiently switch between such configurations. A configuration here refers to a set of software, licenses, drivers, operating system, peripherals, etc. representing the environment of a specific user in a VR system.

In this article we show that recent virtualization technologies are efficient enough to be used as a viable solution to the problem of sharing VR systems and that they offer benefits in terms of automation of management and maintenance operations.

In the task of sharing a VR system, a good degree of flexibility is necessary to switch between configurations. Currently, flexibility can be enforced by using one of the following techniques: 1) HDD switching, that consists in preparing a user configuration as a set of hard-drives and physically installing them in the host machine(s) of the VR system whenever needed; 2) Multiboot that consists of installing different configurations on the VR system machine(s) and selecting one during the boot process; and 3) Deployment of virtual hard drives images that consists of creating hard-drive images files corresponding to each user configuration that can be deployed (i.e. copied) onto the VR system via network communication. These techniques are relatively easy to implement, offer sufficient level of data privacy and ensure optimal performance by directly running on the hardware of VR system host machine(s). However, some drawbacks remain, either on the necessity of physical HDD switching, on the copying of hard drive images, and in all cases in the difficulty to maintain and administrate such solutions.

2 VIRTUALIZATION FOR VR SYSTEMS SHARING

An alternative to classical sharing techniques relies on the use of *virtualization* technologies. To the best of our knowledge, virtualization has not yet been employed for VR systems although it presents interesting benefits.

Virtualization is a technology that aims at abstracting the physical hardware of a host machine by using a virtual machine monitor (also known as a hypervisor). The hypervisor, a minimalist OS, handles access to the physical resources of the hardware across different guest operating systems (also called Virtual Machines - VMs) running on the host. In the context of VR systems, each VM will correspond to a user configuration (i.e. OS, software, drivers, 3D data, etc.). The VMs are totally isolated from the hardware of the host machine and as a consequence have no direct access to the CPUs, GPUs or I/O functionalities. In practice, the hypervisor acts as a controller between the hardware and the VM, and can emulate peripherals or provide means to offer a controlled direct access to hardware components to ensure performance.

Virtualization has the potential to offer a flexible solution for sharing a VR system by simplifying maintenance operations and providing automated features such as batch administration (update, etc.), or copying a configuration in background. This technology also allows for testing or preparing configurations, i.e. creating new VMs, outside of the VR system thus optimizing its uptime. But due to the complexity of VR setups, virtualization has not been evaluated in the context of VR sharing so far.

3 PERFORMANCE EVALUATION

In order to evaluate the efficiency of virtualizing a VR system, we compare results obtained from a native vs. a virtualized mode. The native mode corresponds to a classical installation on which benchmarking applications are directly executed. In the latter mode, a hypervisor controls a virtualized operating system (i.e. a VM) on which the benchmarking applications are executed. Tests were run on a HP Z800 computer with the following components: CPU Intel Xeon E5640@2.67 GHz; 3.7 GB RAM DDR3@1333 MHz; HDD Seagate 500 GB@7200 rpm and with two GPU configurations: #1 NVIDIA Quadro FX 3800 and #2 ATI FirePro V8800. In the virtualized mode, we selected the Xen [1] hypervisor, which offers at the date of this publication, better performances and best hardware compatibility.

3.1 Hardware performance

We first focus on individually evaluating performance of hardware components classically involved in VR setups (USB, networking, RAM, VR peripherals and 3D graphics). Table 1 summarizes the performance measured in native vs. virtualized modes.

*e-mail:jean-marie.normand@ec-nantes.fr
†e-mail:aurelien.milliat@clarte.asso.fr
‡e-mail: marc.christie@irisa.fr
§e-mail:fabien.picarougne@univ-nantes.fr
¶e-mail:guillaume.moreau@ec-nantes.fr

Component	Metric	Native	Virtualized	Gain
USB 2.0	bit-rate	314 Mbps	327 Mbps	+3.97%
RJ45 Gigabit	latency	<1 ms	<1 ms	0.00%
(reception)	bit-rate	292 Mbps	265 Mbps	-9.24%
(emission)	bit-rate	281.9 Mbps	281.3 Mbps	-0.21%
RAM	latency	30.17 ns	31.79 ns	-5.09%
(reading)	bit-rate	3526 MBps	3424 MBps	-2.89%
(writing)	bit-rate	2968 MBps	2859 MBps	-4.00%
HDD	bit-rate	3.14 MBps	3.23 MBps	+2.78%

Table 1: Average performance (over 10 runs) of the main components involved in a VR system. Gain for virtualized mode is expressed in percentage.

While some differences appear in the RAM latency and Gigabit reception, at the practical level the impact is negligible since the benchmark tools (PassMark Advanced Memory Test, PassMark Advanced Network Test) perform successive copies of large memory chunks, or network packets, situations rarely encountered in practice in VR applications. Moreover, in practice, VR peripherals using network communication (cf. Section3.2) did not suffer from noticeable performance degradation.

3.2 Graphics performance

This second study focuses on graphics capacities: dual display, stereoscopy, synchronization and on graphics performance in rendering complex 3D scenes. Table 2 compares graphic chipsets (ATI vs. NVIDIA), the Mono or Stereo display as well as the DirectX9 (DX9), DirectX11 (DX11) or OpenGL drivers. The benchmarking applications mentioned in the table are: **Heaven**: Unigine Heaven Benchmark 4.0, **Valley**: Unigine Valley Benchmark 4.0, and **Furmark**: Geeks3D Furmark v1.9.2. ATI cards have not been tested in stereo due to driver incompatibilities between native and virtualized modes. Applications were run in fullscreen mode at the resolution of 1650×1050 with $8\times$ anti aliasing activated.

Configuration	Native	Virt.	Gain
NVIDIA,Mono, Valley, DX11	338	348	-2,87%
NVIDIA,Mono,Heaven, DX11	229	222	3,15%
NVIDIA,Mono, Valley, OpenGL	276	282	-2,13%
NVIDIA,Mono, Heaven, OpenGL	189	187	1,07%
NVIDIA,Mono, Valley, DX9	376	381	-1,31%
NVIDIA,Mono, Heaven, DX11	278	274	1,46%
NVIDIA,Mono, Furmark, OpenGL	303	311	-2,57%
NVIDIA, Stereo, Heaven, DX11	172	169	1,78%
NVIDIA, Stereo, Valley, DX9	312	288	8,33%
ATI, Mono, Valley, DX11	1088	1098	-0,91%
ATI, Mono, Heaven, DX11	395	398	-0,75%
ATI, Mono, Furmark, OpenGL	1966	1967	-0,05%
ATI, Mono, Valley, OpenGL	495	498	-0,60%
ATI, Mono, Heaven, OpenGL	292	294	-0,68%
ATI, Mono, Valley, DX9	1338	1355	-1,25%
ATI, Mono, Heaven, DX9	901	900	0,11%

Table 2: Comparison of graphical performance between native and virtualized 3D applications on 3D benchmarks in units of the benchmark. Gain for virtualized mode is expressed in percentage.

Differences between monoscopic and stereoscopic modes are limited to a few percent (maximum loss of -2.87%), with however an unexpected gain of 8.33% in the favour of virtualization on the NVIDIA stereo benchmark with DX9. Cases where the virtualized mode was found more efficient is probably due to re-ordering of operations done by the hypervisor. The very clear outcome of these tests is that purely graphics performance are comparable in virtualized and native modes.

Table 3 highlights performance in VR contexts that are measured in frames per second using Fraps software. Two VR applications have been tested: **Improov v1.0** based on Virtools 5.0.0.14 with VRPack 2.6 connected via DTrack to an ART motion capture system, and **Unity 4.1.3** using Middle VR v.1.2.1.

Configuration	Native	Virt.	Gain
ATI, side by side, Unity	106,10	58,29	-82,01%
ATI, master, mono, sync, Improov	43,57	34,78	-25,26%
ATI, slave, mono, sync, Improov	43,58	34,74	-25,46%
NVIDIA, master, mono, Improov	19,77	19,75	-0,06%
NVIDIA, master, stereo, Improov	20,50	21,06	2,66%
NVIDIA, slave, mono, Improov	19,78	19,70	-0,43%
NVIDIA, slave, stereo, Improov	20,46	19,82	-3,24%
NVIDIA, stereo, Unity	168,85	154,15	-9,53%

Table 3: Comparison of graphical performance between native and virtualized VR applications in frames per second and percentage in gain of virtualized applications wrt. native applications.

In a VR context, results demonstrate close to equivalent performance for NVIDIA cards on the two VR applications (with a peak loss of 10% for Unity in stereo). Further investigations are necessary to understand the important performance impact suffered by ATI cards (from 25 to 80%), possibly due to a combination of drivers, hypervisor and virtualized operating system.

Together with these results, compatibility tests have been performed on classical peripherals used in VR: ART tracker (using DTrack1, DTrack2, TrackPack and DTrack2 with TrackPack) connected via RJ45, Virtuose arm 6D35-45 connected via RJ45, 6D mouse from 3D Connexion (USB) and MOTU Audio 3.6.7.0 (Firewire) soundcard. In all cases the peripherals displayed the same behaviour in native and virtualized modes demonstrating an unnoticeable impact of virtualization technologies on these peripherals as well as connection types.

The virtualized solution has been successfully tested on two 4-sided CAVE systems, and has raised significant interest from the maintainers and operators in charge of these platforms.

4 Discussion

Easily sharing and maintaining complex VR setups appear in the short-term future as mandatory steps to offer multiple users with specific needs (companies, laboratories, individuals) a better access to VR technologies, as well as to absorb the costs of such high-end systems. In this paper we showed that, among other techniques, virtualization is an interesting solution for VR system sharing. Indeed, as well as offering close to native performance, both in terms of graphics and of hardware components, this technology present some benefits: backing up a user configuration, batch administration to manipulate user VMs, configuring and testing user configurations outside of the VR room thus increasing availability of the VR room.

Nevertheless, setting up a virtualization solution requires evaluating a chain of hardware and software compatibilities, which can be a time-consuming task. However, the very fast evolution of hardware in the direction of better support of virtualization technologies opens great perspectives in simplifying this task.

References

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. *SIGOPS Oper. Syst. Rev.*, 37(5):164–177, Oct. 2003. 1